Secure Programming in C++

🕘 Duration: 3 Days
Available Languages: English German

Audience

Software Engineers and related roles (developers, architects, security professionals, quality assurance engineers) that create software in C++ for sensitive environments

Precondition

Good knowledge of the C++ Programming Language and basic software development concepts.

Goals

Learn how to write secure code in C++.

Contents

- Generic Secure Programming Knowledge
  - # CVE - Common Vulnerabilities and Exposures
  - # CWE - Common Weakness Enumeration
  - # Impact of insecure programming on security and functional safety
  - # Principle of Least Privilege
- Attack Vectors to protect against
  - # Compromised Credentials
  - # Weak Credentials
  - # Insider Threats
  - # Missing or Poor Encryption
  - # Misconfiguration
  - # Buffer overflows/overruns/underflows/underruns
  - # NOP slides and Padding
  - # Junk data and Junk code
  - # Heap Spraying
  - # Path Traversal
- Secure Programming and the SDLC
  - # Requirements Engineering
  - # Security Target of Evaluation
  - # Testing and Test-Driven Development
  - # Continuous Integration
  - # Continuous Design Improvement
  - # Git Flow vs Trunk-Based Development
  - # Pair and Ensemble Programming
  - # Code Reviews and Code Walkthroughs

- Rules and Recommendations
  - # The C Preprocessor
  - # Declarations and Initialization
  - # Expressions
  - # Integers
  - # Floating Point
  - # Arrays
  - # Containers
  - # Characters and Strings
  - # Memory Management
  - # Input and Output
  - # Exceptions and Error Handling
  - # Object-Oriented Programming
  - # Environment
  - # Signals
  - # Error Handling
  - # Application Programming Interfaces
  - # Concurrency
  - # Miscellaneous
  - # POSIX / Linux / Mac OS
  - # Microsoft Windows
- Side-Channel Attacks and Software Defenses
  - # Flow Control
  - # Redundant Masked Parameters
  - # Time-invariant algorithms
  - # Memory Checksums
  - # Watermarks
  - # Noise Injection
- Using Static Code Analysis to find Security Issues
  - # PC-Lint

The course is largely, but not exclusively, based on the SEI CERT C and C++ Coding Standards. The course uses C++23 with the GNU C Compiler or Clang.

The course language is C++. Nelkinda also offers this course in other languages, for example, C, Java, and Kotlin.

Note

The exercises for the PC-Lint section are only available for participants that have a Gimpel PC-Lint license.

Event Type

This is a full-day instructor-led open (anyone can register) or in-house classroom training about Secure Programming in C++. The course comprises of live lecture/presentation, interactive instructor-led live coding, and instructor-guided hands-on pair/ensemble labs and exercises. The number of seats is limited to ensure the best quality training for the participants. For open training, the course fee includes snacks and lunch.

Trainer

Your trainer for this event is Christian Hujer.

Christian Hujer has experience with embedded CPU and Microcontrollers since 1984, for example, Zilog Z80A, MOS 6502, Motorola, 68000, Samsung CalmRISC/SecuCalm, ARM, Infineon TriCore, Atmel AVR, Hitachi H8, and Intel 80x51. He has 20 years of experience in secure programming. He's been training developers and teams for organizations like BNP Paribas, Elsevier, Giesecke & Devrient, Nokia, SUN Microsystems, Volkswagen, and many others.

Booking

Contact Siddhesh Nikude, +91-95-52572354, training@nelkinda.com